De la predicţia salturilor condiţionate la o problemă ştiinţifică fundamentală şi… dincolo de ea

Curs festiv, Promoţia "Calculatoare"
16 mai, 2008

Prof. Lucian VINŢAN

Catedra de calculatoare, Universitatea "Lucian Blaga",
Str. Emil Cioran, No. 4, 550025 Sibiu, Romania
lucian.vintan@ulbsibiu.ro
http://webspace.ulbsibiu.ro/lucian.vintan/

Scurtă recapitulare: *branch prediction*

- **Necesitatea** predicţiei dinamice adaptive
- Predictoare ***state of the art*** (markoviene, neurale, hibride)
- Câteva **probleme actuale**:
  - Salturi/apeluri indirecte registru (OOP, *virtual machine systems; Intel Pentium M*)
  - 28% LD_miss/BRANCH (LVP, addr_correlation, SMT…)
  - ***Branch*-uri nepolarizate având comportare "aleatoare"?** (1999-Milano, 2006-China, 2007-Korea)

Ce sunt *branch*-urile nepolarizate?

$$P(S_i) = \max(f_0,\ f_1) = \begin{cases} f_0, & f_0 \geq 0.5 \\ f_1, & f_0 < 0.5 \end{cases}$$

- $S=\{S_1, S_2, \ldots, S_k\}$ = set of distinct contexts that appear during all branch instances;
- *k* = number of distinct contexts, $k \leq 2^p$, where *p* is the length of the binary context;
- $f_0 = \frac{T}{T+NT}$, $f_1 = \frac{NT}{T+NT}$, and obviously $f_0 + f_1 = 1$;
- if $P(S_i)=1,\ (\forall) i=1, 2, \ldots, k,$ then the context S_i is **completely biased** (100%);
- if $P(S_i)=0.5,\ (\forall) i=1, 2, \ldots, k,$ then the context S_i is **totally unbiased**.

Ce e un... *Shuffled Dynamic Branch*?

$$D(S_i) = \begin{cases} 0, & n_t = 0 \\ \dfrac{n_t}{2 \cdot \min(NT, T)}, & n_t > 0 \end{cases}$$

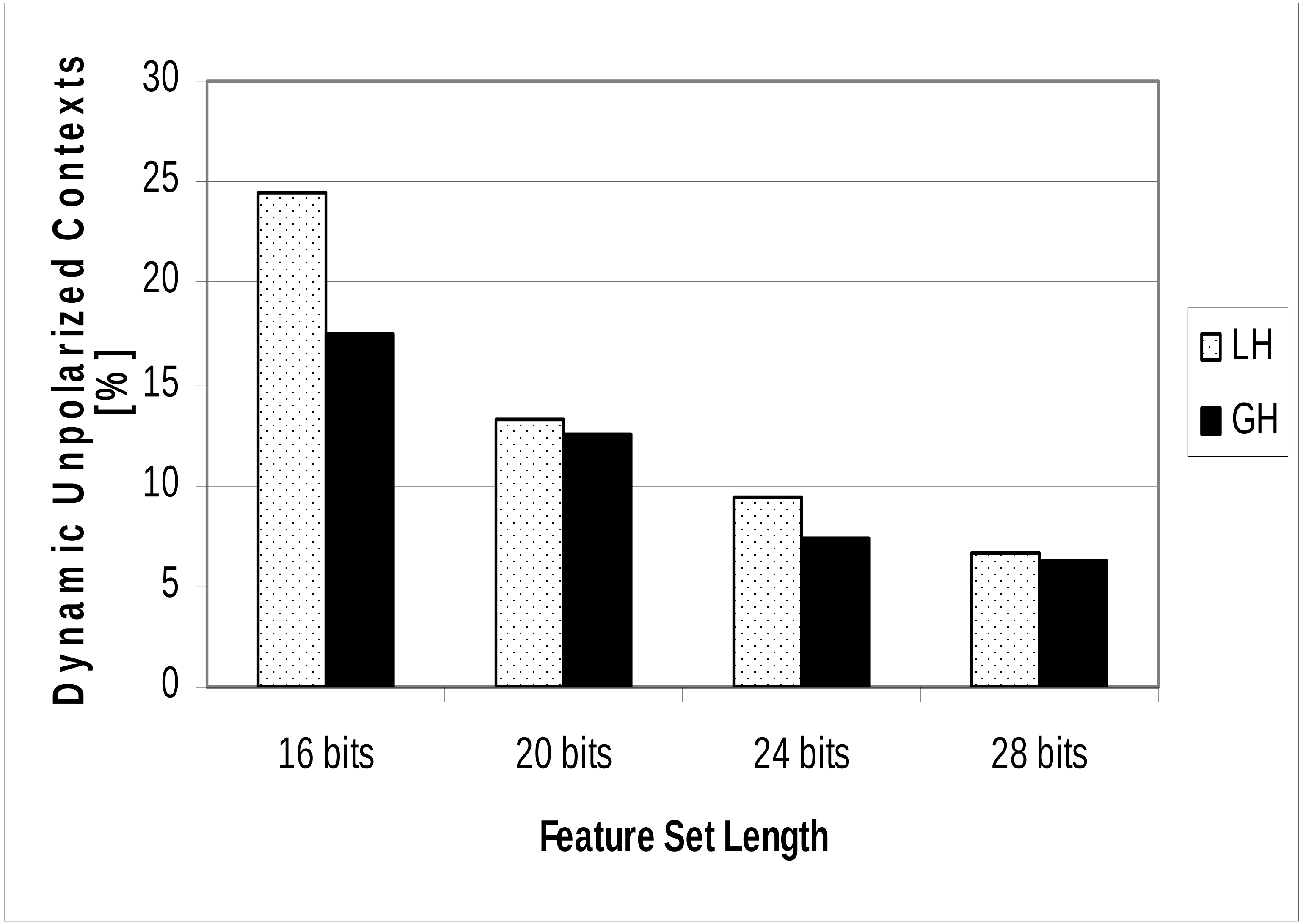
- n_t = the number of branch outcome transitions, from (0→1 or 1→0), in a certain context *Si*;
- $2 \cdot \min(NT, T)$ = maximum number of possible transitions;
- if $D(S_i) \to 1, (\forall) i = 1, 2, ..., k$, then the behavior of the branch is **totally shuffled**;
- if $D(S_i) \to 0, (\forall) i = 1, 2, ..., k$, then the behavior of the branch in context *Si* is constant.

O clasă de *branch*-uri impredictibile?

- Hard-to-Predict Dynamic Branch= [Unbiased_Branch] AND [Shuffled Branch]

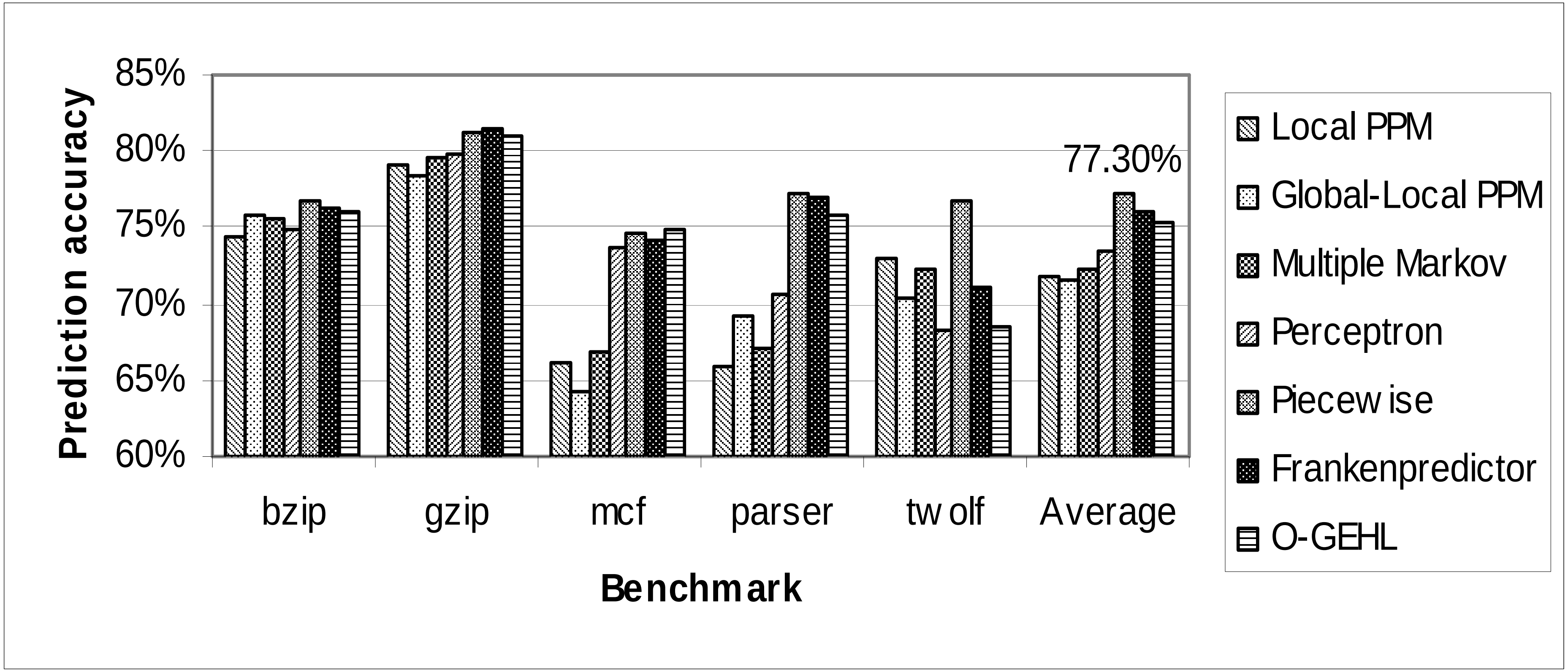
- [P(Si)<0.95] AND [D(Si)>0.2] → Prediction accuracy<~75%

Reducerea nr. de *unbiased branches* prin extensia contextului

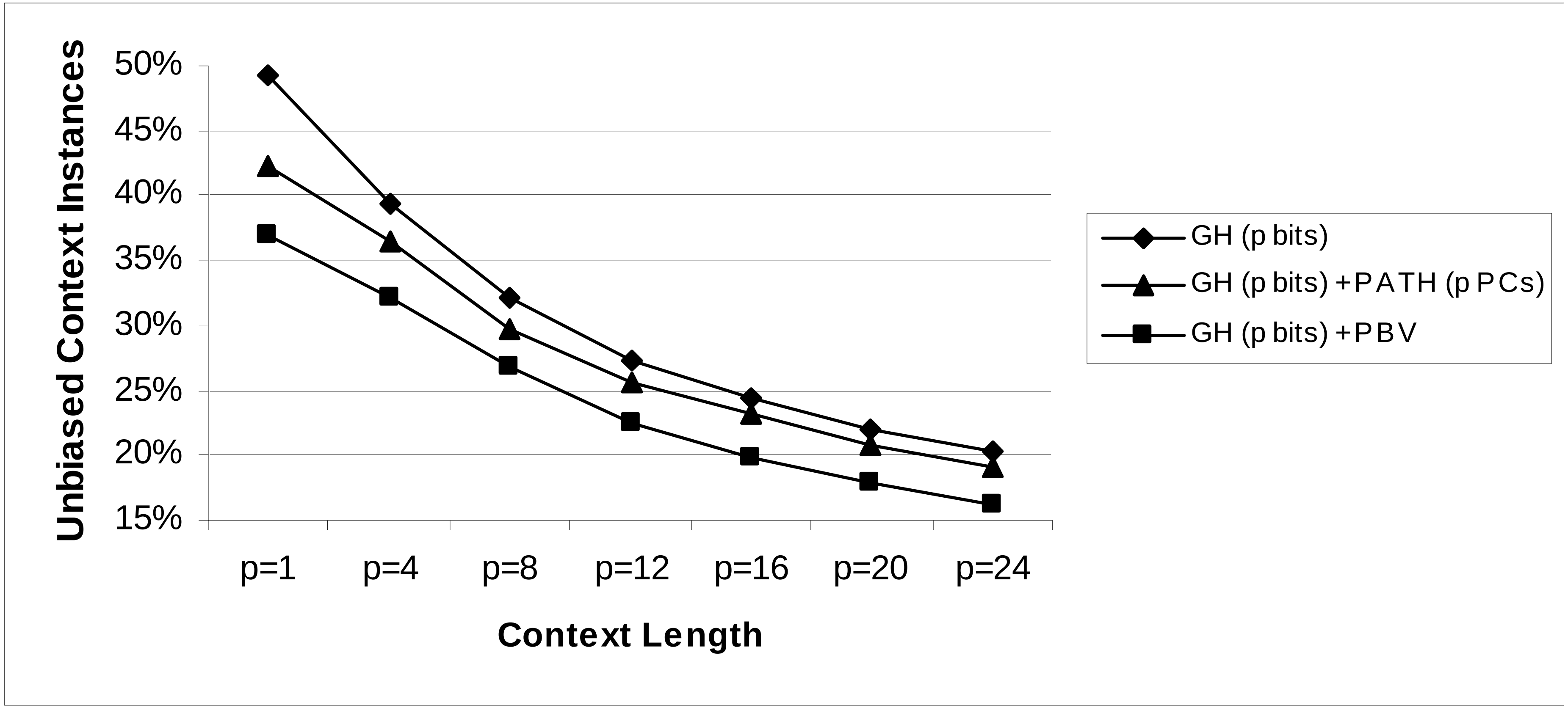


Prediction accuracies obtained using state-of-the-art branch predictors

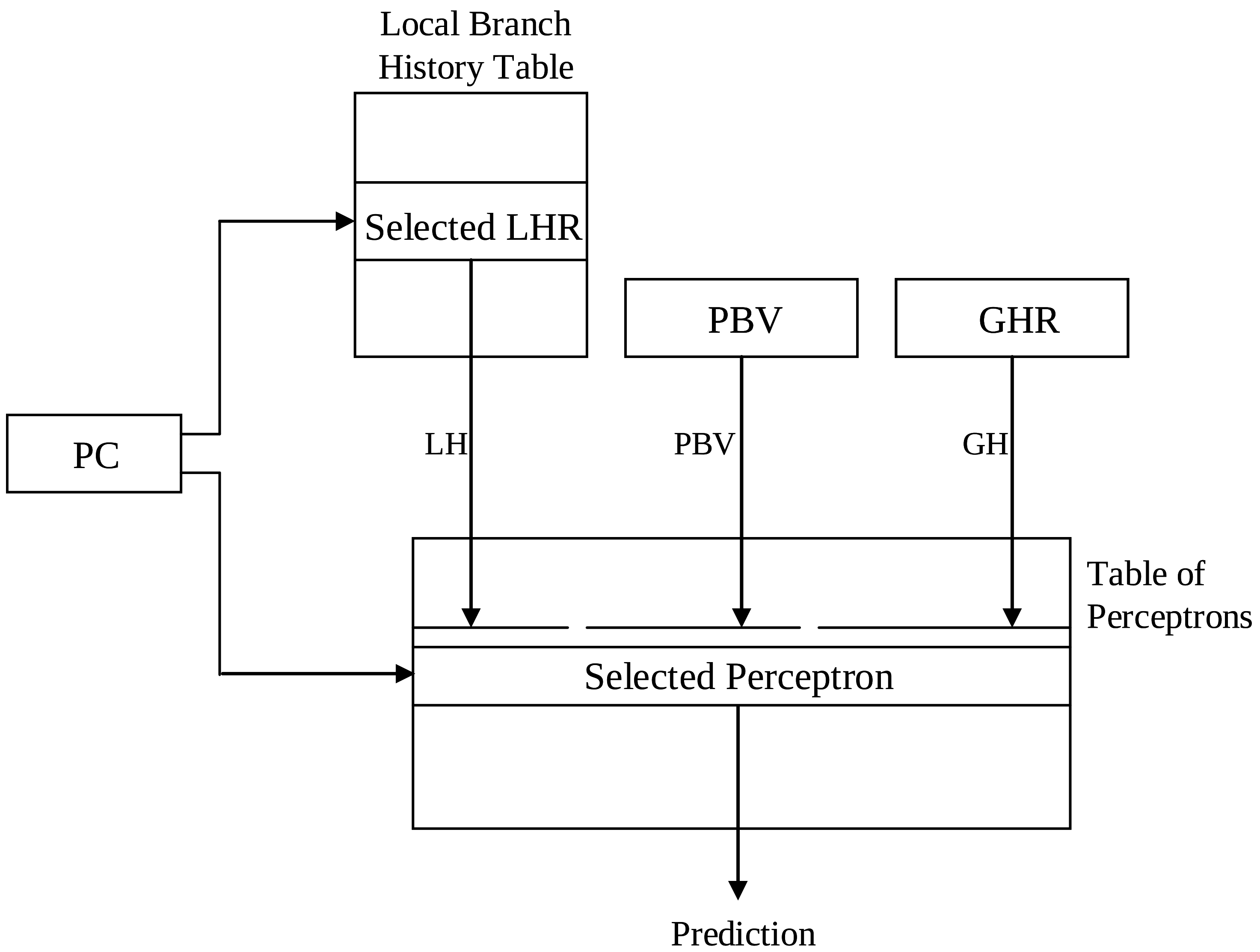
INTEL Championship Branch Prediction Competition (CBP-2), Orlando, Florida, USA, (2006), http://camino.rutgers.edu/cbp2/



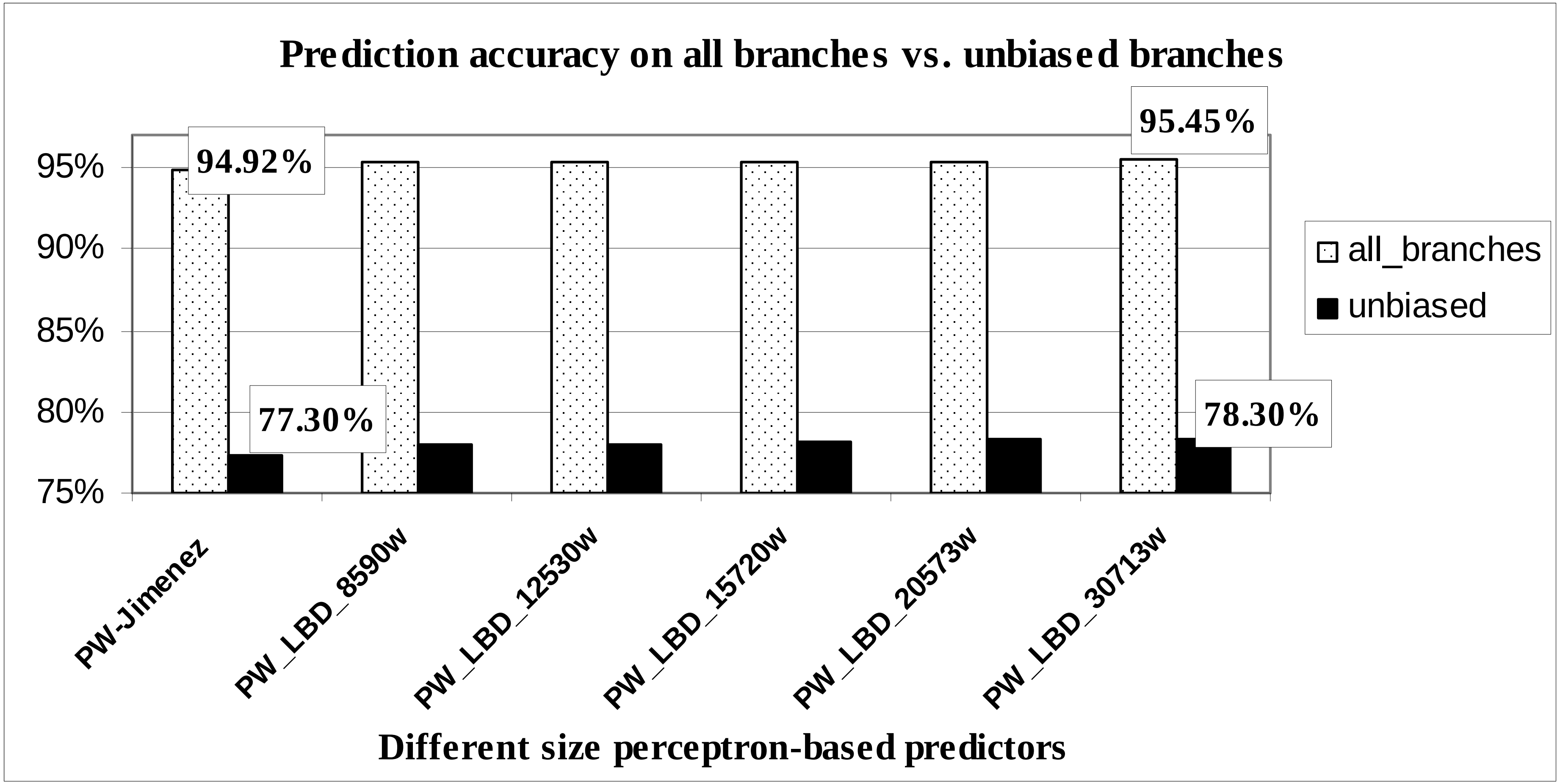
Reduction of average percentages of unbiased branches by adding new information (*PATH*, PBV)



Piecewise linear branch predictor using the previous branch's value



Piecewise linear branch predictors: results



Ce e o secvenţă “aleatoare”?

- **Ce înseamnă aleator?**
- În particular, ce înseamnă un **şir binar aleator**?
  - **010101010101010101010...**
  - **011010100000100111100 11...**
- Se poate da o **definiţie matematică intrinsecă, ontologică, a unui şir aleator** de simboluri?
- Poate genera **rularea unui program determinist secvenţe "aleatoare"?**
- Pentru şiruri de simboluri nu se poate defini decât **gradul de aleatorism**, diferenţa între aleator şi non-aleator fiind deci una **vagă**.

Definiţia aleatorului după **Borel**

- Pentru orice *n*, **blocurile de *n* termeni din şir au aceeaşi probabilitate** P de apariţie în şir. Alfabet de *m* simboluri → P = $\mathbf{1/m^n}$.
- Aproape toate numerele reale, scrise în reprezentare zecimală infinită, sunt aleatoare.
- Definiţia lui Borel **nu este constructivă** (efectivă), în sensul permiterii generării de şiruri aleatoare.
- **[Problemă deschisă**: constantele matematice transcendente sunt Borel-normale?]

Maşina **Turing** redivivus!

- $f: Q \times S \rightarrow Q \times S \times M$
- $TM_t(x) = \{q(t), s(t); q(t+1), s(t+1), m\}$, $m \in M = \{L, R\}$ // **Instructiune**
- $Q = \{q_0 q_1 q_2 \ldots q_f\}$, $s \in S = \{0, 1, \mathit{blank}\}$

- Secvenţa de intrare *x* aparţine **mulţimii tuturor secvenţelor binare finite** (FB – *Finite Binary*).

- O TM defineşte **o funcţie parţială a mulţimii FB** pe ea însăşi → **algoritm=procesare de simboluri** pe o TM.

Conjectura **Church-Turing**

- **Pentru orice algoritm** (procedură care se termină într-un număr finit de paşi) **există o maşină Turing echivalentă.** → *Algoritm=TM*

- Noţiunea de **algoritm** impune ca acesta **să poată fi implementat pe o maşină Turing**. Cu alte cuvinte, dacă un calculator poate implementa un algoritm acesta poate fi implementat şi de o anumită maşină Turing.

Mulţimea tuturor maşinilor Turing este numărabilă

- Definiţia noţiunii de **numărabil; Paradoxuri** ale infiniţilor actuali.
- **Justificare:** se poate trece de la mulţimea maşinilor Turing cu *k* instrucţiuni (TM^k) la cea a celor cu (*k+1*) instrucţiuni (TM^{k+1}), $\forall k$ =3,4,5... → TM^k este numărabilă. Evident că în cadrul unei mulţimi TM^k există un număr finit de maşini Turing → **mulţimea tuturor TM este numărabilă**, **Q.E.D.**

Funcţiile Turing-calculabile sunt "infinit-puţine"!

- O funcţie parţială f:N→N este „**Turing-calculabilă**" dacă $\forall n \in N, \exists x \in FB$ cu n=c(x) pentru care maşina se opreşte generând la final codificarea binară a lui n prin funcţia c: f(n)=c(TM(x)).
- Cum mulţimea TM este numărabilă → **mulţimea funcţiilor parţiale Turing-calculabile este şi ea numărabilă**. Această mulţime aparţine **mulţimii <u>tuturor</u> funcţiilor parţiale f:N→N**, care este **nenumărabilă** (fiind practic izomorfă cu mulţimea numerelor reale).
- Rezultă că funcţiile calculabile sunt infinit-puţine (cardinal *alef 0*) → **Funcţiile non-calculabile sunt infinit-majoritare!**

Aleatorul e majoritar dar nu-l putem exprima!

- Orice **şir binar aleator**, NU ar trebui să fie generat printr-o **funcţie parţiala Turing-calculabilă.** (Există secvenţe necalculabile de simboluri care nu sunt, totuşi, aleatoare!)
- Asociind secvenţele aleatoare cu funcţiile parţiale care nu sunt Turing-calculabile → aceste **secvenţe aleatoare sunt nenumărabile** (de puterea continuului), deci **infinit-majoritare** pe mulţimea tuturor funcţiilor parţiale f:N→N.
- Aşadar: **secvenţele deterministe sunt numărabile, cele aleatoare sunt nenumărabile!**
- Deşi riguroasă, definiţia nu e efectivă → **eşec practic!**

Definiţii neriguroase, dar mai practice, ale gradului de aleatorism

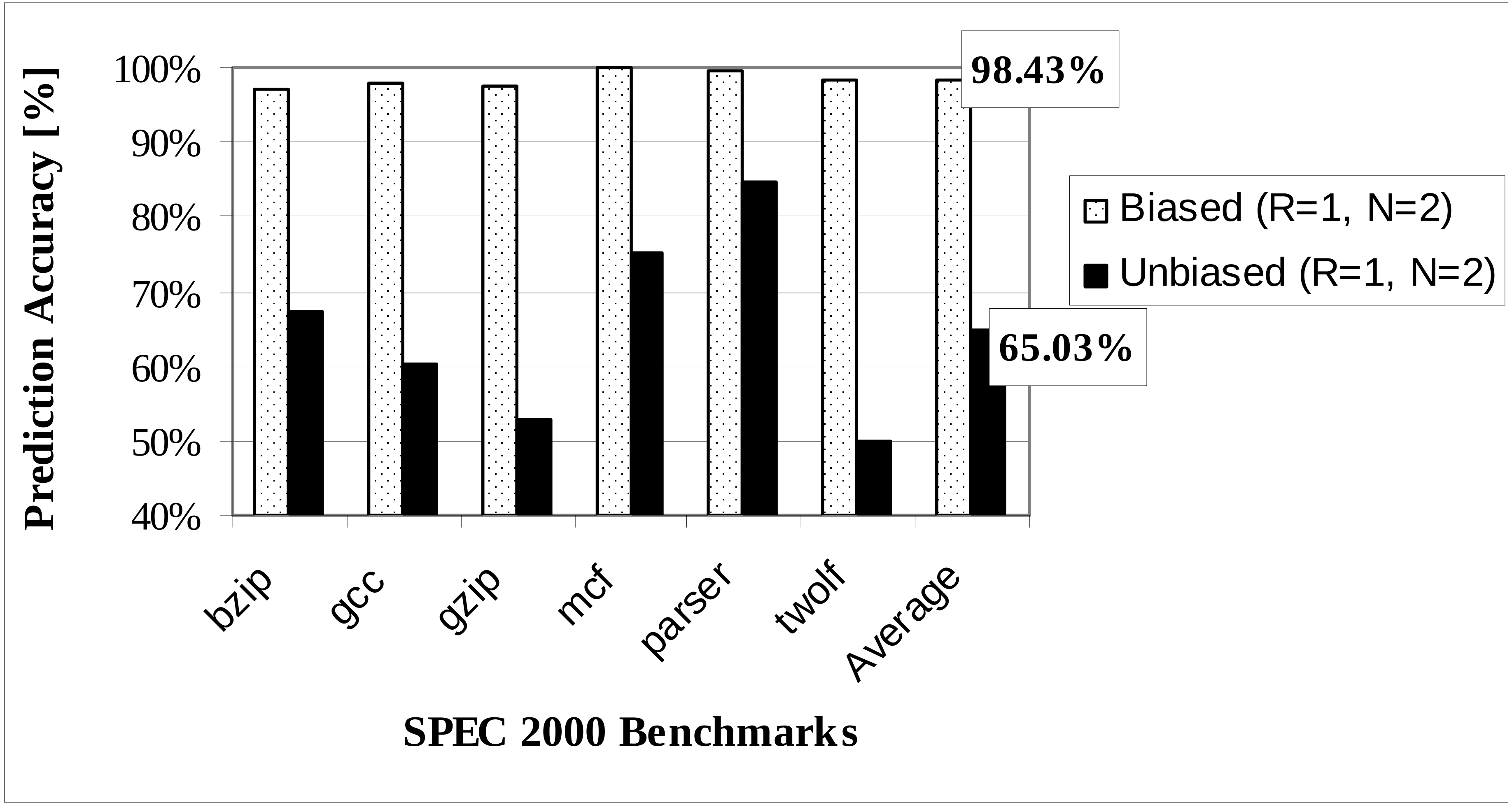
Bazate pe:

- **Predictibilitatea *Hidden Markov Models*** (HMM, *L. Rabiner*)
- **Entropia discretă** a secvenţei
- **Rata de compresie** a secvenţei (*Huffman, Gzip* etc.). O secventă aleatoare e incompresibilă.
- **Complexitatea *Kolmogorov*** a codului generator al secvenţei de simboluri

Predictibilitatea HMM – o limită ultimativă?

- A **HMM** is a doubly embedded stochastic process with a hidden stochastic process that can only be observed through another stochastic process - that generate the sequence of observable symbols.
- Our **hypothesis**: HMMs could compensate **relevant information miss-knowledge** through its underlying stochastic process that is not observable.

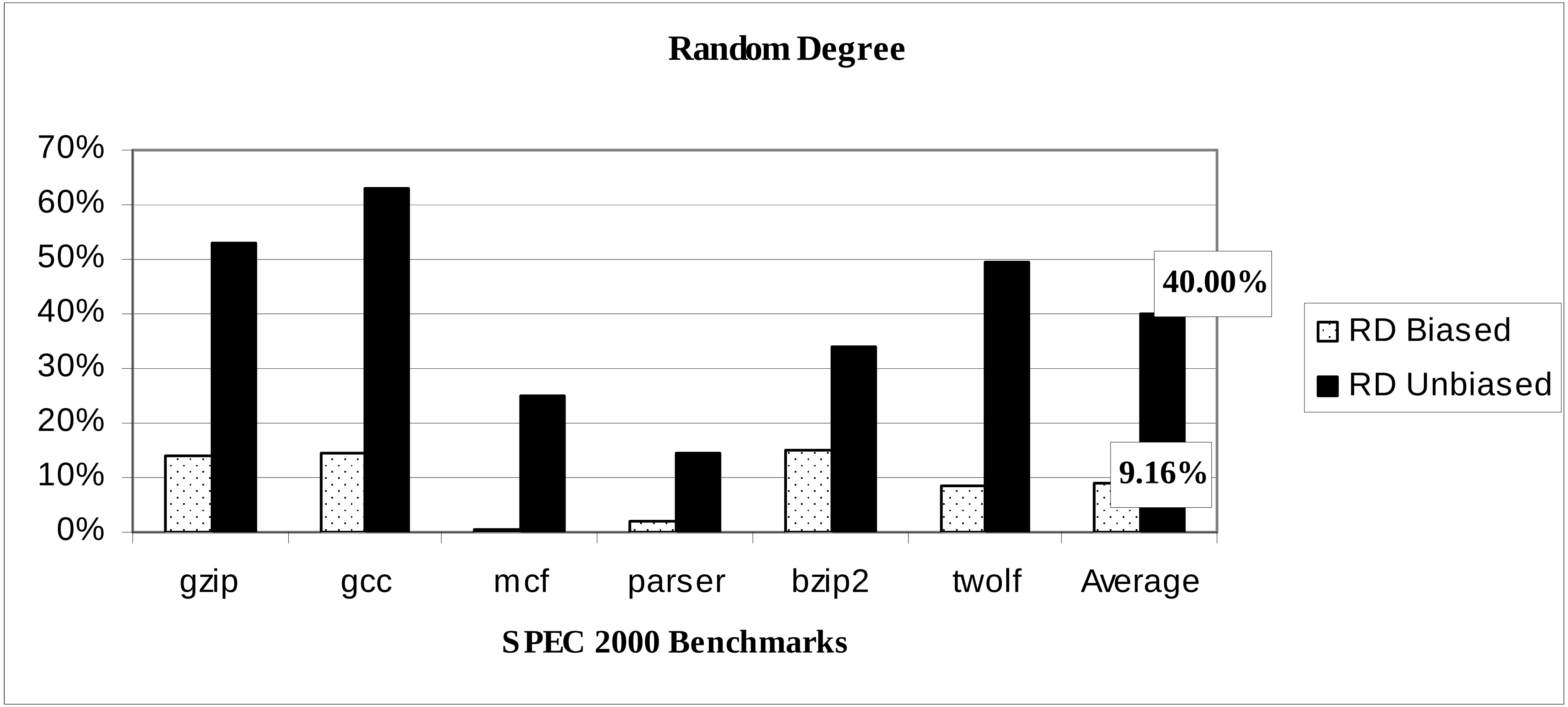
Predicţia "celui mai performant HMM" (R=1, N=2)



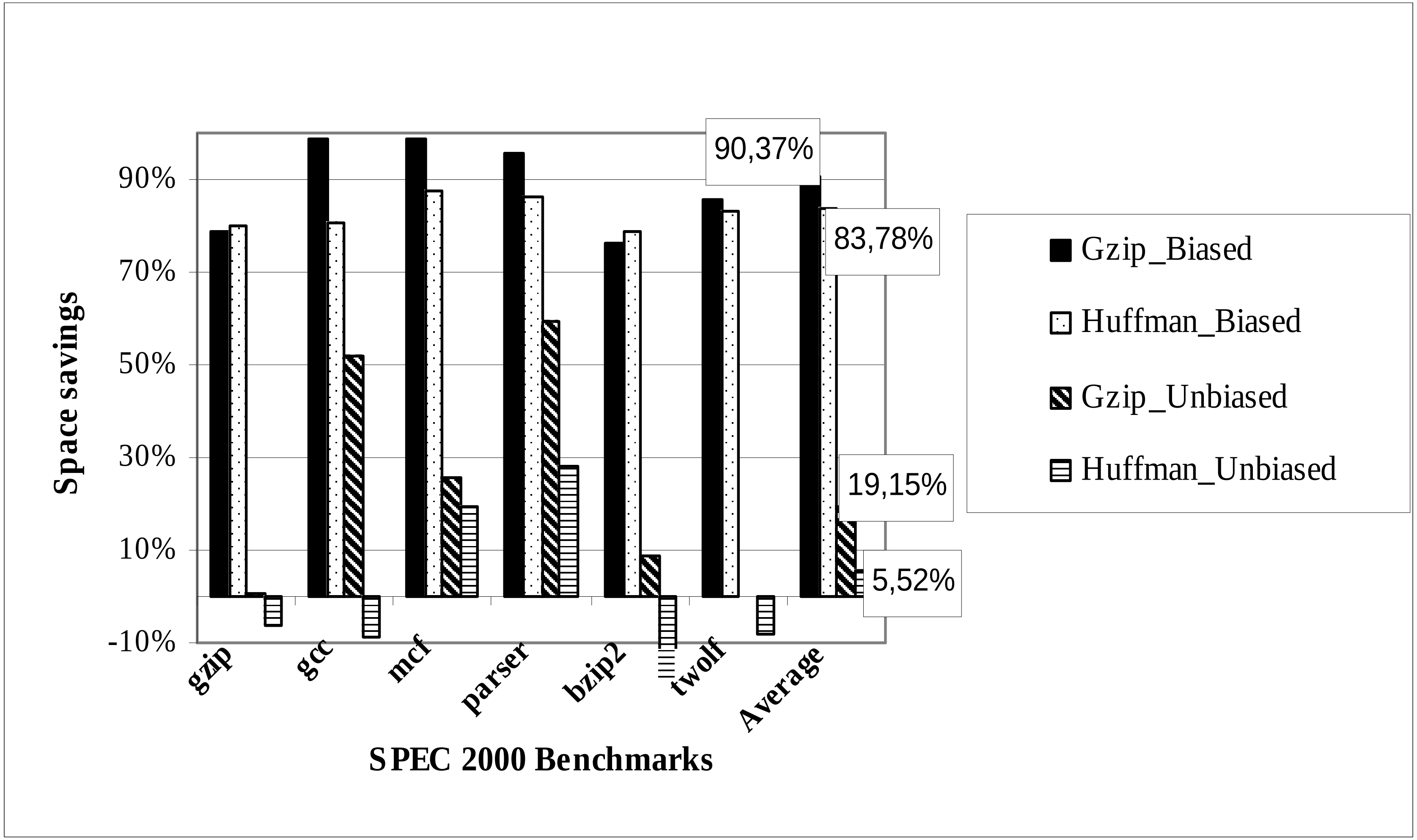
Entropia discretă a secvenţei

- $E(S) = -\sum_{i=1}^{k} P(Xi) \log P(Xi) \geq 0$
- $D(S_i) = \begin{cases} 0, & n_t = 0 \\ \dfrac{n_t}{2 \cdot \min(NT, T)}, & n_t > 0 \end{cases}$
- $RD(S) = D(S) \cdot E(S) \in [0, \log_2 k]$
- Max_RD(Binary_string)=1 (100%, complet aleator)
- 01010101010101... maximizes both D and E but…

Gradul de aleatorism: *biased vs. unbiased branches*



Rata de compresie a secvenţei



"Complexitatea *Kolmogorov*" a codului generator

```
Permute (int n){
  int k;
  pctr = pctr+1;
  if(n != 1) # the first branch instruction analyzed (PC=35) BIASED (predictibil)
  {
    Permute(n-1);
     for( k = n-1; k >= 1; k--)# the second analyzed branch (PC=58) UNBIASED!
    {
            Swap(&permarray[n], &permarray[k]);
            Permute(n-1);
            Swap(&permarray[n], &permarray[k]);
    }
  }
}
```

***K*(Br_58)** = Minimum 42 HSA instructions or 8 C instructions.
***K*(Br_35)** = 12 HSA instructions or 3 C instructions

→ *K*(Br_58)> *K*(Br_35)

Concluzii(I)

- **Nu există o paradigmă eficientă** pt. aleatorismul unei secvenţe (şir).
- Predicţia eficientă a *branch*-urilor nepolarizate: o **problemă deschisă** (spaţiu de grad superior, convenabil reprezentării problemei → **predictor eficient**)
- **Gradele de aleatorism intrinsec -** de mare ajutor pentru ansamblul arhitectură-compilator (***predictor improvement***). Gradul de haos determinist.

Concluzii(II)

- **Creşterea complexităţii** proiectelor (*object oriented programs, design patterns, complex projects management,* etc.) → creşterea nr. de *unbiased branches*
- Influenţa acestor *branch*-uri în arhitecturile ***multi-thread*** si ***many-core*** e o problemă importantă si deschisa.
- **Arhitectura calculatoarelor** nu poate fi înţeleasa profund fără ajutorul teoriei <u>algoritmilor</u> (calculabilităţii), teoriei <u>informaţiei</u>, teoriei <u>proceselor stohastice</u>, <u>învăţării automate</u> etc.

Concluzii(III)

- Acest curs a prezentat o **aventură eşuată,** dpdv utilitar!
- Care este **scopul ultimativ** al ŞTIINŢEI şi al TEHNOLOGIEI?
  - Utilitar?
  - **Cognitiv?**
  - Estetic?! Etc.
  - Scopul ultimativ, esential, al stiintei il constituie **onoarea spiritului uman**!